On the Solution of Min-Max Problems in Robust Optimization

Massimiliano Vasile

Department of Mechanical & Aerospace Engineering,
University of Strathclyde. 75 Montrose Street, G1 1XJ Glasgow, UK

Abstract. The paper presents a simple memetic algorithm for the solution of min-max problems. It will be shown how some of the heuristics provide the analogous mechanism of other evolutionary and non-evolutionary heuristics proposed in the literature. It will be also argued that some existing heuristics might not be sufficient to correctly solve the problem and to avoid the so called *red-queen effect*.

1 Introduction

Nowadays the prediction of the performance of a system is essentially based on computer simulations. The uncertainty in the computer model and input parameters, however, implies that the predicted behavior of the system can have a considerable variability. The worst case scenario is defined as the condition in which the performance of the system is the worst possible among all those who can be predicted with current information. Designing the system to be optimal in the worst case scenario is the most conservative but also the most robust solution and provide a lower limit on the achievable performance of the system. The optimization of the worst case scenario translates into the solution of a global min-max problem in which the worst case is defined as a maximum in the uncertain space and the optimal design corresponds to the minimum of all the maxima in the design space. This paper presents an evolutionary-based algorithm for the solution of min-max problems coming from worst-case scenario optimization. Past examples of the use of evolutionary optimization can be found in [1, 6] that combined evolutionary computation and surrogate modelling. Other solution methods in the literature are based on more deterministic approaches in combination with surrogate modelling, like in [2]. This paper presents a memetic strategy that intentionally does not make use of surrogate models. The goal of the paper is to demonstrate the ability of the proposed strategy to consistently identify the global solution of the min-max problem. The paper starts with a general formulation of the min-max problem addressed in this paper. Section 3 then introduces the proposed algorithm and related heuristics. Section 4 presents the test benchmark while Section 5 shows some preliminary results.

2 Min-Max Problem

The problem of interest can be formulated as follows:

$$\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d}, \mathbf{u}) \tag{1}$$

where $\mathbf{d}$ is a vector of design (or control) variables defined in the design space D, $\mathbf{u}$ is a vector of uncertain variables, defined in the uncertain space U, and f is the performance index of the system or process defined by the combination of the uncertain and design parameters. It is important to underline that in the general case, both the minimization over D and the maximization over U are global optimization problems and that f is multimodal and can be non differentiable everywhere in U and D.

3 Solution Algorithms

The algorithm proposed in this paper is a combination of the one proposed in [3] and [2]. In [2] it is proposed to iteratively solve the following two problems, one after the other:

$$\mathbf{u}_a = \text{argmax}_{\mathbf{u}\in U} f(\mathbf{d}_{min}, \mathbf{u}) \tag{2}$$

$$\mathbf{d}_{min} = \text{argmin}_{\mathbf{d}\in D}\{\max_{\mathbf{u}_a \in A_u} f(\mathbf{d}, \mathbf{u}_a)\} \tag{3}$$

where the archive A_u is a collection of all the $\mathbf{u}_a$ generated by the solution of problem (2) for each new $\mathbf{d}_{min}$ generated by the solution of problem (3). Problem (2) can be seen as a restoration of the maximum condition on U, therefore the whole process can be considered as a *minimization-restoration* loop.

It is important, at this point to observe that, if a population-based method is used to solve problem (3), the subproblem $\max_{\mathbf{u}_a \in A_u} f(\mathbf{d}, \mathbf{u}_a)$ can be interpreted as a cross-check of the $\mathbf{u}$ associated to a population P of $\mathbf{d}$ values as in [3]. For each $\mathbf{d}$, in fact, problem (3) requires selecting the $\mathbf{u}_{a,max}$ that maximizes f among all the $\mathbf{u}_a$ found thus far. This principle is equivalent to the game theoretic Nash ascendancy relationship used in [5], as it corresponds to selecting the best strategy, among all the ones that the players in the population P can play. In the scheme proposed by [2], however, the minimization over D is performed assuming that the elements in the archive are not updated while f is minimized over D. The main consequence is that convergence can be achieved if the archive A_u contains a sufficient number of elements. In [3] instead the solutions were recalculated either running a global or a local optimization as $\mathbf{d}$ was changing. The main reason for the latter strategy is that a variation of $\mathbf{d}$, seen from the space U, can correspond to a change in the location of the maxima. When this occurs the solution of problem (3) can lead to values of $\mathbf{d}_{min}$ such that the maximum of f over A_u is very far even from a local optimum. The whole process, therefore, might iterate for a long time between minimization and restoration without converging. This is what can be called *red queen effect*.

Here, it is proposed to solve both problems (2) and (3) with Inflationary Differential Evolution [4] and to allow the algorithm to compute for each $\mathbf{d}$ a local maximum $\mathbf{u}_a^*$ starting from each element in A_u. The value $\mathbf{d}_{min}$ with associated local maximum $\mathbf{u}_{a,max}^* = \text{argmax}_{\mathbf{u}_a^* \in A_u^*} f(\mathbf{d}_{min}, \mathbf{u}_a^*)$, are then saved in the archive A_d and the elements in the archive A_d are cross-checked to maximize the change to identify the global maximum in U. The overall strategy is presented in Algorithm 1.

Algorithm 1 IDEA$minmax$

Initialize $\bar{\mathbf{d}}$ at random and run $\mathbf{u}_a = \text{argmax}_{\mathbf{u}\in U} f(\bar{\mathbf{d}}, \mathbf{u})$
$A_u = A_u \bigcup \{\mathbf{u}_a\}$
while $n_{feval} < n_{feval,max}$ **do**
  Run $\mathbf{d}_{min} = \text{argmin}_{\mathbf{d}\in D}\{\max_{\mathbf{u}_a^*\in A_u^*} f(\mathbf{d}, \mathbf{u}_a^*)\}$
  Run $\mathbf{u}_a = \text{argmax}_{\mathbf{u}\in U} f(\mathbf{d}_{min}, \mathbf{u})$
  if $f(\mathbf{d}_{min}, \mathbf{u}_{a,max}^*) < f(\mathbf{d}_{min}, \mathbf{u}_a)$ **then**
    $A_u = A_u \bigcup \{\mathbf{u}_a\}$, $A_d = A_d \bigcup \{\mathbf{d}_{min}, \mathbf{u}_a\}$
  else
    $A_u = A_u \bigcup \{\mathbf{u}_{a,max}^*\}$, $A_d = A_d \bigcup \{\mathbf{d}_{min}, \mathbf{u}_{a,max}^*\}$
  end if
end while
Run Cross Check Algorithm 3 over the archive A_d

Algorithm 2 $\max_{\mathbf{u}_a^*\in A_u^*} f(\mathbf{d}, \mathbf{u}_a^*)$

for all the elements in A_u **do**
  Run local search from $\mathbf{u}_a \in A_u$ and compute $\mathbf{u}_a^* = \text{argmax}_{\mathbf{u}\in U} f(\mathbf{d}_{min}, \mathbf{u})$
  Add local maximum to the set of local maxima $A_u^* = A_u^* \bigcup \{\mathbf{u}_a^*\}$
end for
$\mathbf{u}_{a,max}^* = \text{argmax}_{\mathbf{u}_a^*\in A_u^*} f(\mathbf{d}_{min}, \mathbf{u}_a^*)$

4 Test Benchmark

In order to test the algorithm presented in the previous section we use the same test benchmark of 13 toy problems proposed in [2] and we add 4 more that have a higher level of complexity. The four additional test cases are in Table 1 where n represents the number of dimensions in D and U (i.e. the total size of the problem is $2n$).

Table 1. Min-Max Test Benchmark

ID	Function	Boundaries
EM1	$f(\mathbf{d}, \mathbf{u}) = \sum_i ((u_i - 3d_i)\sin u_i + (d_i - 2)^2)$	$\mathbf{d} \in [0, 2\pi]^n, \mathbf{u} \in [0, 20]^n$
MV8	$f(\mathbf{d}, \mathbf{u}) = \sum_i^n ((2\pi - u_i)\cos(u_i - d_i) - u_i \sin u_i + 0.1d_i)$	$\mathbf{d} \in [-5, 2]^n, \mathbf{u} \in [0, 2\pi]^n$
MV9	$f(\mathbf{d}, \mathbf{u}) = \sum_i^n ((d_i - u_i)(\cos(-5u_i + 3d_i)))$	$\mathbf{d} \in [-5, 2]^n, \mathbf{u} \in [0, 2\pi]^n$
MV11	$f(\mathbf{d}, \mathbf{u}) = \sum_i^n (-10d_i \sqrt{\text{abs}(\cos(d_i u_i))} + u_i + 5(d_i - 5)^2)$	$\mathbf{d} \in [1, 9]^n, \mathbf{u} \in [-2, 2]^n$

5 Some Results

The results in Table 2 show the total number of function evaluations required to reach the same, or better, accuracy than Ref. [2] on the 13 benchmark problems. The values

Algorithm 3 Cross Check

```
Initialize Δ, tol_Δ
while Δ > tol_Δ do
    for all the elements in A_d do
        Compute local maximum f(d_i, u_j^*) from u_j ∈ A_d
        Δ = f(d_i, u_j^*) − f(d_i, u_i)
        if Δ > tol_Δ then
            u_i = u_j^*
        end if
    end for
end while
```

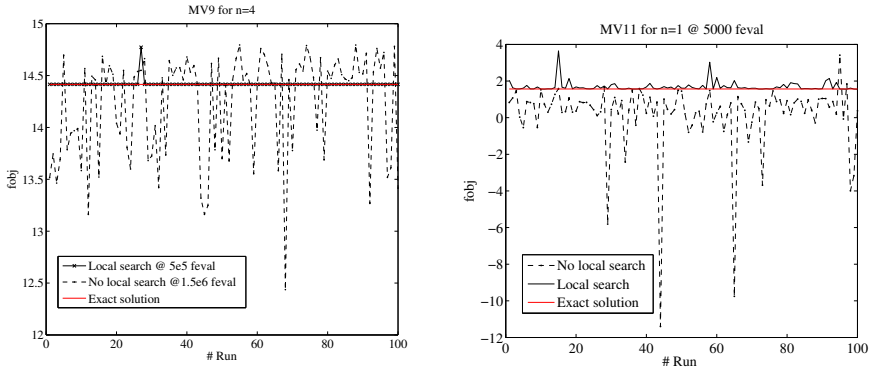
are the mean and standard deviation (over 100 runs) of the difference between the exact solution and the solution found by the algorithm proposed in this paper. It is worth reminding that no surrogate is used in this paper, hence the tests are only assessing the ability of the evolutionary process in Algorithms 1 and 2 to converge. It is also interesting to note that in all the 13 cases convergence is achieved without the need for tracking the local maxima as the number of local maxima is limited and the basins of attraction are large. On the other hand, the results obtained with Algorithm 1 tend to be more conservative than the ones in Ref. [2], because the algorithm in Ref. [2] finds solutions that are not always maximising f in U. The number of function calls is between 3 and 40 times higher than the one in Ref. [2], due to the absence of the surrogate model, and between 10 and 200 times lower than the one in [5]. It has to be said, however, that the quality of the result depends on the ratio of the function evaluations allocated to the optimization in D and in U. At the same time, previous work by the author [6] has shown how the use of surrogates can provide up to 98% of reduction in function calls, which is in line with the results in [2]. Figure1(a) shows the

Table 2. Min-Max Test Benchmark [2]

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
nfeval	5000	5000	15000	7000	4000	50000	90000	500	4000	5000	5000	2000	10000
mean	4e-8	1e-4	2e-5	6e-4	3e-5	5e-7	3e-3	1e-17	0.0	2e-16	7e-5	4e-16	8e-5
std	2e-7	9e-4	1e-6	1e-3	5e-5	9e-7	3e-3	1e-16	0.0	2e-15	3e-4	2e-15	2e-4

outcome of 100 runs of Algorithm 1, with and without local search, applied to MV9 for $n = 4$. The test with no local search was run for a total of 1.5e6 function evaluations, while the test with local search was run for a total of 5e5 function evaluations. The continuous line is the exact solution with value $f^* = 14.415$. This results demonstrates that if the number of local maxima is very high and the basins of attraction are small then the archive A_u would need to be filled with a very high number of samples to provide good convergence. In other words, looking for the maximum f over the values in the archive A_u, for each value of $\mathbf{d}$, is equivalent to a random sampling drawing

samples from the distribution of the local maxima found so far. If $\mathbf{d}$ is changing the location of the maxima then random sampling could be insufficient (unless the number of samples is very high). On the other hand, if for every $\mathbf{d}$ one was searching for the maximum of the local maxima, then the process would be equivalent to a multistart local search in which the starting points are drawn from the distribution of the local maxima found so far. In this later case, although the search in U is significantly more expensive, convergence can be achieved for a lower number of elements in the archive A_u. All this is even more true if the size of the basins of attraction of the minima in D



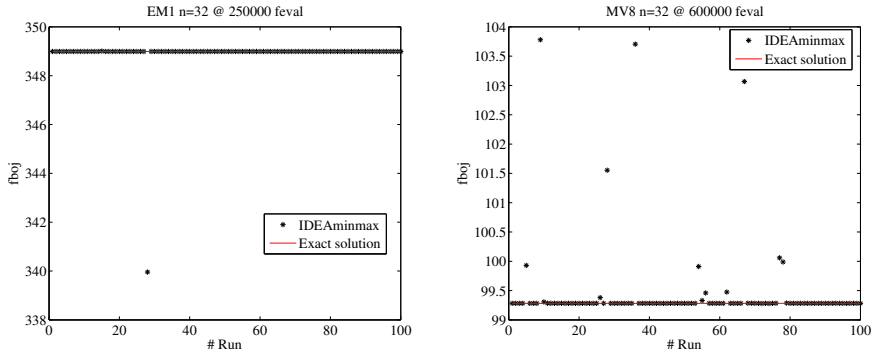
(a) (b)

Fig. 1. Results over 100 runs:(a) function MV9 for n=4 (b) function MV11 for n=1.

is significantly larger than the size of the basins of attraction of the maxima as in the case of function MV11. Figure1(b) shows the result of the application of Algorithm1 to the solution of problem MV11 for $n = 1$, The reference solution, marked with a red line, has value $f^* = 1.5659$. In the case the number of local maxima is low or $\mathbf{d}$ does not change the location of the maxima, then the use of the simple function values is sufficient as can be seen if Figures 2 where Algorithm 1, with no local search, is applied to the solution of problems EM1 and MV8 for $n = 32$. The reference solution for EM1 has value $f^* = 348.9897$ and for MV8 has value $f^* = 99.2825$.

6 Conclusions

The memetic algorithm proposed in this paper was shown to efficiently and consistently provide solutions to global min-max single objective problems with up to 64 dimensions. In some cases it was shown that a search in the space of the local maxima is required to avoid the red queen effect and converge to a globally optimal solution. The algorithm does not use any surrogate model to further reduce the computational cost, however it was demonstrated, in previous work by the author, that a gain up to 98% is potentially achievable.



(a) (b)

Fig. 2. Results over 100 runs:(a)function EM1 for n=32 (b) function MV8 for n=32.

References

1. A. Zhou and Q. Zhang, "A Surrogate-Assisted Evolutionary Algorithm for Minimax Optimization," *2010 IEEE Conference on Evolutionary Computation (CEC 2010)*, Jul. 2010.
2. J. Marzat, E. Walker and H. Piet-Lahanier, "Worst-case Global Optimization of Black-Box Functions through Kriging and Relaxation," *J. Global Optim.* vol. 55, pp. 707-727, 2013.
3. M. Vasile, E. Minisci and Q. Wijnands, "Approximated Computation of Belief Functions for Robust Design Optimization," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Apr. 2012.
4. M. Vasile, E. Minisci and M. Locatelli, "An Inflationary Differential Evolution Algorithm for Space Trajectory Optimization," *Trans. Evol. Comp.* vol. 15, pp. 267-281, Apr. 2011.
5. R. I. Lung, D. Dumitrescu, "A New Evolutionary Approach to Minimax Problems",2011 IEEE Congress on Evolutionary Computation (CEC),5-8 June 2011, New Orleans, US, pp. 1902-1905, 10.1109/CEC.2011.5949847.
6. S. Alicino, M. Vasile "Surrogate-based Maximisation of Belief Function for Robust Design Optimisation". 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 8-11, 2013, DOI: 10.2514/6.2013-1757.